**TASK-2**

**1.Load the rest countries data using html and js scripts file and run a loop on the data and print all country name in console.**

**Sol-**

**Index.html file-**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<script src='script.js'></script>

</body>

</html>

**Script.js file-**

var request=new XMLHttpRequest();//1.create a new request

request.open('GET','https://restcountries.eu/rest/v2/all',true);//2.create a connection

request.send();//3.send teh connection

request.onload=function()//4.register event listener to load the data as and when available

{

var countrydata=JSON.parse(this.response);

var elem=0;

for( var elem in countrydata)

{

console.log(countrydata[elem].name);

}

};

**2.Write a write up on copy by value and copy by reference**

Copy by value is also called as deep copy and copy by reference is also called as shallow copy. The major differences are as below,

|  |  |
| --- | --- |
| **Shallow Copy** | **Deep Copy** |
| Process of constructing a new collection of object and then populating it with reference to the child objects found in the original. | Process of constructing a new collection of object and then recursively populating it copies of the child object found in the original. |
| Does not recurse and it does not create copies of child objects themselves. | Creates a complete independent clone of the original object and all its children. |
| Shallow copy is not recursive. | Deep copy is recursive. |
| No new memory will be allocated instead refernce will eb made to the original data. | New memory allocation will be done to the cloned copy. |

**3.Write a write up on copy by value a composite datatype(arrays+objects)**

Objects and strings come under composite data types. They work on concept of copy by value meaning, whenever a reference to any of the object say for eg., array-a[] is made by some other array-b[], then any changes made to any of the array’s would be reflecting in all the other corresponding references made to the changed array. Same is the case with an objects created in javascript. For composite data types, no separate memory allocation takes place, instead references are made whenever an object/duplicate is created as needed.